# 学习报告（git）

1813003 李昱龙 0615

1.了解git这个软件

Git是什么？

Git是目前世界上最先进的分布式版本控制系统（没有之一）。

Git有什么特点？简单来说就是：高端大气上档次！

经过一天的了解，git确实能在日常工作中给予程序员很大的帮助，无论是个人操作还是团队协作，很幸运能不工作在手动管理多个“版本”的史前时代，哈哈哈。Git的优势不单是不必联网这么简单， Git极其强大的分支管理，把SVN等远远抛在了后面。

2.git的安装初步使用和连接github

在git中创建属于自己的版本库，这里可以学到的有 git init，听从廖雪峰的建议下载了notepad++来写文本文档。编写文档，放入git可以用到git add git commit

即：

1.使用命令git add <file>，注意，可反复多次使用，添加多个文件；

2.使用命令git commit -m <message>，完成。

3.要随时掌握工作区的状态，使用git status命令。

4.如果git status告诉你有文件被修改过，用git diff可以查看修改内容。

那我要是想要看看自己都做了什么怎么操作呢？

5.用git log可以查看提交历史

6.要重返未来，用git reflog查看命令历史，以便确定要回到未来的哪个版本。（我还可以回去的~）

那我还是不太理解git工作是怎么实现的怎么办？

偷几张图哈哈哈





7.阿这，写错了怎么办，还手贱点了提交？

第一种：用命令git checkout -- file。

第二种：第一步用命令git reset HEAD <file>，就回到了第一种，第二步按一操作。

版本回退！

8.看文件不顺眼想删掉！

rm 之后 git rm 或者不删了 git checkout --<filename>

9.不搞github？

那不可能！

ssh-keygen -t rsa -C "youremail@example.com" 创建SSh key然后一路回车，在github的add SSHkey中输入秘钥，再create a new repository填名字后create就搞好了，仓库就有了~用git push -u origin master把我们master的内容推上去。

10.clone？

git clone 加地址就ok了~

3.分支管理~

1.首先咱得有: git branch <name>

查看分支：git branch

切换分支：git checkout <name>或者git switch <name>

创建+切换分支：git checkout -b <name>或者git switch -c <name>

合并某分支到当前分支：git merge <name>

删除分支：git branch -d <name>

2.合并

git merge （Fast forward）

或者

Git merge –no-ff可以看出历史

2.有情况？

git log –graph看看合并图~

修复bug时，我们会通过创建新的bug分支进行修复，然后合并，最后删除；

把工作现场git stash一下，然后去修复bug，修复后，再git stash pop，回到工作现场；

在master分支上修复的bug，想要合并到当前dev分支，可以用git cherry-pick <commit>命令，把bug提交的修改“复制”到当前分支，避免重复劳动。

3.卸磨杀驴（狗头）

git branch -D <name>强删

4.重头戏来了

多人协作~

git remote来查看远程库的信息或者更详尽？git remote -v

git push origin master（名字）//推送分支

5.小结：

建立本地分支和远程分支的关联，使用git branch --set-upstream branch-name origin/branch-name；

从本地推送分支，使用git push origin branch-name，如果推送失败，先用git pull抓取远程的新提交；

在本地创建和远程分支对应的分支，使用git checkout -b branch-name origin/branch-name，本地和远程分支的名称最好一致；

从远程抓取分支，使用git pull，如果有冲突，要先处理冲突

4.标签：

建一个！git tag <name>

用命令git tag -a <tagname> -m "blablabla..."指定信息。

建完看看：用命令git show <tagname>可以看到说明文字：

不想用了？

命令git tag -d <tagname>可以删除一个本地标签；

命令git push origin :refs/tags/<tagname>可以删除一个远程标签。

5.git自定义~

1.忽略某些文件时，需要编写.gitignore；

2. git config --global alias.<别名> ‘现名’

6.用SourceTree

一句话，可视化这么好，真香

需要进一步搞一搞的！

Git服务器搭建没整明白

协作还是没上手，有的懵